|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №** 3

«САПР ПЛИС. Язык описания аппаратуры»

по дисциплине

«Архитектура вычислительных машин и систем»

|  |  |
| --- | --- |
| Выполнил студент группы  ИВБО-01-22 | Зырянов М.А |
| Принял ассистент кафедры ВТ | Дуксина И.И. |
| Практическая работа выполнена | « \_\_ » \_\_\_\_\_\_\_ 2023 г. |
| «Зачтено» | « \_\_ » \_\_\_\_\_\_\_ 2023 г. |

Москва 2023

АННОТАЦИЯ

Данная работа включает в себя 1 рисунок, 1 таблицу, 2 формулы, 1 листинг. Количество страниц в работе — 15.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc152679803)

[1 ПРЕОБРАЗОВАНИЕ ЛОГИЧЕСКОЙ ФУНКЦИИ 5](#_Toc152679804)

[2 ПОСТРОЕНИЕ МОДУЛЕЙ 7](#_Toc152679805)

[3 ИСХОДНЫЙ КОД 9](#_Toc152679806)

[4 СКРИНШОТЫ ВРЕМЕННЫХ ДИАГРАММ 12](#_Toc152679807)

[ЗАКЛЮЧЕНИЕ 13](#_Toc152679808)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc152679809)

Введение

Для логической функции, заданной в векторном виде, необходимо восстановить таблицу истинности. Создать модуль на языке Verilog HDL, описывающий реализацию логической функции в СДНФ. Создать модуль на языке Verilog HDL, описывающий реализацию логической функции в МДНФ.

# 1 преобразование логической функции

Исходя из персонального варианта практической работы №1 имеется Логическая функция в векторном виде: F049BD7E

Используя простейший алгоритм перевода шестнадцатиричного числа в его двоичное представление было получено число: 11110000010010011011110101111110

По данному числу была воссоздана Таблица 1.1 таблица истинности :

Таблица 1.1 — Таблица истинности

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | x5 | F |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |

Продолжение Таблицы 1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

# 2 ПОСТРОЕНИЕ МОДУЛЕЙ

Для начала выпишем МДНФ функции (Формула 2.1):

(2.1)

Далее выпишем СДНФ функции (Формула 2.2)

(2.2)

Создадим модули на языке Verilog HDL, описывающее реализацию логической функции в СДНФ, МНДФ, а также тестирующий модуль. Для этого создадим модуль с именем «sdnf», «mdnf». Также, создадим модуль с именем «testbench».

Модуль СДНФ получает на вход сигнал in, состоящий из 5 бит, номер старшего разряда равен 4. Сигнал отвечает за 5 логических переменных, от которых зависит данная функция. Модуль возвращает выходной сигнал f, которому присваивается значение логического выражения, которое записывается с помощью операторов логического ИЛИ, И, побитовой инверсии.

Модуль МДНФ создается аналогичным образом. Модуль получает на вход сигнал in, состоящий из 5 бит, номер старшего разряда равен 4. Сигнал отвечает за 5 логических переменных, от которых зависит данная функция. Модуль возвращает выходной сигнал f, которому присваивается значение логического выражения, которое записывается с помощью операторов логического ИЛИ,И, побитовой инверсии.

Тестирующий модуль проверяет корректность работы реализованных модулей СДНФ и МДНФ. В начале алгоритма объявляется 6-битный регистр args и 32-битные регистры reference\_reg, error\_reg1, error\_reg2 со старшим разрядом 0 и младшим 31, а также цепи clk, reg1, reg2. Регистру reference\_reg присваивается значение исходной функции. Регистру args и остальным цепям присваивается значение 0. Каждые 10 миллисекунд при изменении сигнала с 0 на 1 осуществляется поэлементное заполнение регистров error\_reg1, error\_reg2. Цепям reg1, reg2 присваиваются выходные значения модулей СДНФ и МДНФ. Регистры ошибок, содержащие значения ffffff после отработки модулей, сигнализируют о корректности их работы.

# 3 ИСХОДНЫЙ КОД

Листинг 3.1 – Модуль, описывающий реализацию СДНФ

|  |
| --- |
| *`timescale 1ns / 1ps*  *module sdnf (*  *input wire [4:0] in,*  *output wire f*  *);*  *assign f = (~in[4] && ~in[3] && ~in[2] && ~in[1] && ~in[0]) ||*  *(~in[4] && ~in[3] && ~in[2] && ~in[1] && in[0]) ||*  *(~in[4] && ~in[3] && ~in[2] && in[1] && ~in[0]) ||*  *(~in[4] && ~in[3] && ~in[2] && in[1] && in[0]) ||*  *(~in[4] && in[3] && ~in[2] && ~in[1] && in[0]) ||*  *(~in[4] && in[3] && in[2] && ~in[1] && ~in[0]) ||*  *(~in[4] && in[3] && in[2] && in[1] && in[0]) ||*  *(in[4] && ~in[3] && ~in[2] && ~in[1] && ~in[0]) ||*  *(in[4] && ~in[3] && ~in[2] && in[1] && ~in[0]) ||*  *(in[4] && ~in[3] && ~in[2] && in[1] && in[0]) ||*  *(in[4] && ~in[3] && in[2] && ~in[1] && ~in[0]) ||*  *(in[4] && ~in[3] && in[2] && ~in[1] && in[0]) ||*  *(in[4] && ~in[3] && in[2] && in[1] && in[0]) ||*  *(in[4] && in[3] && ~in[2] && ~in[1] && in[0]) ||*  *(in[4] && in[3] && ~in[2] && in[1] && ~in[0]) ||*  *(in[4] && in[3] && ~in[2] && in[1] && in[0]) ||*  *(in[4] && in[3] && in[2] && ~in[1] && ~in[0]) ||*  *(in[4] && in[3] && in[2] && ~in[1] && in[0])||*  *(in[4] && in[3] && in[2] && in[1] && ~in[0]);*  *endmodule* |

Листинг 3.2 – Модуль, описывающий реализацию МДНФ

|  |
| --- |
| *module mdnf (*  *input wire [4:0] in,*  *output wire f*  *);*  *assign f = (in[3] & in[2] & ~in[1] & ~in[0]) ||*  *(~in[4] & in[3] & in[2] & in[1] & in[0]) ||*  *(~in[4] & ~in[3] & ~in[2]) ||*  *(in[4] & in[2] & ~in[1]) ||*  *(in[4] & ~in[2] & in[1]) ||*  *(~in[3] & ~in[2] & ~in[0]) ||*  *(in[4] & in[3] & in[2] & ~in[0]) ||*  *(in[4] & ~in[3] & in[2] & in[0]) ||*  *(in[3] & ~in[2] & ~in[1] & in[0]);*  *endmodule* |

Листинг 3.3 – Модуль тестирования

|  |
| --- |
| *`timescale 1ns / 1ps*  *module testbench();*  *reg [5:0] args;*  *reg clk;*  *wire reg1, reg2;*  *reg [0:31] reference\_reg, error\_reg1, error\_reg2;*  *initial*  *begin*  *reference\_reg = 32'h8E146896;*  *args = 5'b00000;*  *clk = 0;*  *error\_reg1 = 0;*  *error\_reg2 = 0;*  *end*  *always #10 clk = ~clk;*  *always @(posedge clk)*  *begin*  *error\_reg1[args] = reg1 ~^ reference\_reg[args];*  *error\_reg2[args] = reg2 ~^ reference\_reg[args];*  *args = args + 1;*  *if(args == 6'd33)*  *$finish;*  *end*  *sdnf mod1 (.in(args[4:0]), .f(reg1));*  *mdnf mod2 (.in(args[4:0]), .f(reg2));*  *endmodule* |

# 4 СКРИНШОТЫ ВРЕМЕННЫХ ДИАГРАММ

Используя Vivado получим временную диаграмму работы кода, реализующие СДНФ и МДНФ рассматриваемых функций (Рисунок 4.1).

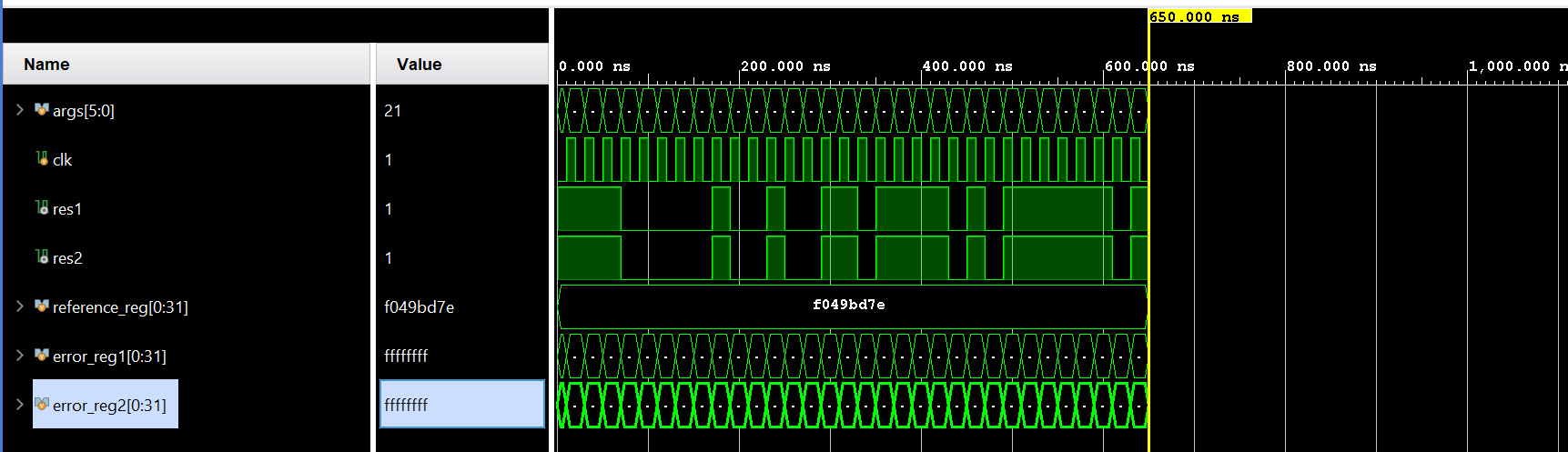


Рисунок 4.1 – Временная диаграмма

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были получены формулы СДНФ и МДНФ заданной функции. В САПР Vivado, согласно полученным формулам, были реализованы модули СДНФ и МДНФ функции, а также проведена их успешная верификация на тестирующем модуле в режиме симуляции.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 1 — URL: https://online-edu.mirea.ru/mod/resource/view.php?id=405132 (Дата обращения: 23.09.2022).

2. Методические указания по ПР № 2 — URL: https://online-edu.mirea.ru/mod/resource/view.php?id=409130 (Дата обращения: 23.09.2022).

3. Смирнов С.С. Информатика [Электронный ресурс]: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов — М., МИРЭА — Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).

4. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.

5. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).

6. Антик М.И. Математическая логика и программирование в логике [Электронный ресурс]: Учебное пособие / Антик М.И., Бражникова Е.В.— М.: МИРЭА – Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).

7. Жемчужникова Т.Н. Конспект лекций по дисциплине «Архитектура вычислительных машин и систем» — URL: https://drive.google.com/file/d/12OAi2\_axJ6mRr4hCbXs-mYs8Kfp4YEfj/view?us  
p=sharing (Дата обращения: 23.09.2022).

8. Антик М.И. Теория автоматов в проектировании цифровых схем [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА – Российский технологический университет, 2020. — 1 электрон. опт. диск (CD-ROM).

9. Орлов С.А. Организация ЭВМ и систем: Учебник для вузов. 4-е изд. — СПб.: Питер, 2018. — 688 с.: ил.

10. Шустов М.А. Цифровая схемотехника. Основы построения. — СПб.: Наука и Техника, 2018. — 320 с.: ил.

11. Рафиков Р. А. Электронные сигналы и цепи. Цифровые сигналы и устройства: Учебное пособие. — СПб.: Издательство «Лань», 2016. — 320 c., ил. — (Учебники для вузов. Специальная литература).

12. Угрюмов Е. П. Цифровая схемотехника: учеб. пособие для вузов. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 816 с.: ил.